UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/542,714	04/04/2000	ALLAN HAVEMOSE	AMI 99 0006	5268

32718 7590 05/22/2003

GATEWAY, INC.
14303 GATEWAY PLACE
ATTENTION: MARK S. WALKER (MAIL DROP SD-21)
POWAY, CA 92064

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2124	6

DATE MAILED: 05/22/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/542,714	HAVEMOSE, ALLAN	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover she t with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 March 2003* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-44* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 April 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This action is responsive to the Applicant's response filed March 20, 2003. As indicated in Applicant's response, claims 1, 7, 13, and 18 have been amended. Claims 1-44 are pending in the office action.

Specification

2. The disclosure is objected to because of the following informalities: the term "compiles" in p. 5, line 7 appears to be an incorrect fit/usage in the sentence and should be replaced with appropriate pronoun or noun.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-5, 7-11, 13-16, 18, and 20-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blandy, USPN: 6,295,642 (art of record, hereinafter Blandy) in view of Coutts et al., USPN: 6,311,165 (hereinafter Coutts).

As per claim 1, Blandy discloses a method for dynamic compiling, such method comprising loading byte code (e.g. *PDA* - col. 3, lines 37-54; *classes loader, Bytecodes* - col. 4, lines 7-16) on a data processing processor, i.e. digital information appliance as claimed;

such byte code suitable for instrumentation such as having a tagged section (*lock* – col. 4, lines 54-65 – Note: use of lock to hold up portions of byte code against dynamic compilation

is equivalent to tagging section of code; *breakpoints* – col. 5, lines 14-23 – Note: breakpoints used to ward off compilation control over uncompiled sections is equivalent to instrumentation tagging);

identifying the tagged section of byte code (e.g. *lock, breakpoint, branch monitor* - col. 4, line 54 to col. 5, line 23; *range, GTR* – Fig 4 – Note: scope and range of bytecodes already compiled are equivalent to identifying tagged portions) and compiling (e.g. Fig. 3; step *414* - Fig. 4) such tagged section;

wherein the tagged section is compiled when loaded (e.g. col. 3, lines 24-28; col. 4, lines 7-16) so as to enable the processor to utilize it without additional compiling (e.g. col. 5, lines 15 to col. 6, line 33; Fig. 3, 4 – Note: selecting a goto point, using fallthru instructions and branch breakpoints to skip over uncompiled sections to generate native code are equivalent to using the already compiled parts of the code while running it through the JIT without additional recompiling – see *will be interpreted* – col. 6, lines 30-33).

But Blandy does not specify that the byte code includes one dynamic base object, such object comprising an interface dynamic object and an implementation dynamic object communicating with each other over a message bus. However, Blandy discloses a communication system having a bus interconnecting appliances, client or host system memory to the peripheral and network interfaces (Fig. 1); a runtime environment using object-oriented loader subsystem (e.g. *runtime data access 204* - Fig. 2); and a development environment including a server downloading to requesting appliances with limited resources; and desire to spare storage use therein (e.g. col. 1, lines 50-61; col. 2, lines 1-5; *PDA* - col. 3, lines 45-54). Further, Coutts, in a method to provide applications and transaction data to peripheral devices,

i.e. appliances, via download and communications similar to the communications environment of Blandy, such application being object-oriented and provided with JIT (e.g col. 25, line 66 to col. 26, line 10) to alleviate the appliance's resource utilization reminiscent of the tagging/skip compiling scheme taught by Blandy, discloses a application environment with a server module with legacy data, module application with interface, and a peer-to-peer messaging module(e.g. Fig. 32). One skilled in the art would recognize that the combination of those is analogous to a dynamic base object having a implementation object (server module), a interface object (application host module), and a message bus linking the 2 objects because the whole architecture thus disclosed by Coutts involves a object-oriented model instantiated into specific objects for users (or digital information appliances) to request for a transaction (e.g. *object models* - col. 27, lines 26-50); such that all those features have been suggested via Blandy's disclosures from above, i.e. a server for providing resources, a requesting appliance and a protocol to communicate the dynamic request as in a web client/server session transaction. It would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the client/server application communication paradigm and byte code load/execute runtime environment as presented by Blandy so that the application code thus downloaded to the appliance would include an dynamic base object as suggested by Coutts with 3 object included therein: a implementation object, a interface object and a messaging object, or message bus as claimed. One skilled in the art would be motivated to do so because first, Blandy's load and runtime JIT compiling scheme entails the need for a dynamic instance of data transacting process on an appliance with limited resources (i.e. obviating the need for such appliance to generate an transaction object using its own resources), thus a need to include a dynamic base object ready in

the downloaded application program; second, Blandy's communication network is an object-oriented base programming application where client/server machines and peripherals are attached by bus and inherent communication protocol, hence a need for a dynamic message object and interface object to communicate the request; and finally because Blandy's appliance wherein a download request is initiated by the peripheral devices or appliances to a server suggests a need for an runtime implementation object storing server's instance of retrieving and assembling all the information or application components related to the appliance's request from a server repository, i.e. a implementation object.

As per claim 2, Blandy further discloses encoding application source code in byte-code (e.g. Fig. 6), such byte-code suitable for further analysis (e.g. step 304 – Fig. 3); and tagging a section (e.g. *breakpoints* – col. 5, lines 14-23).

As per claim 3, Blandy further discloses that the byte-code is suitable for instrumentation (e.g. *branch monitor 416*-Fig. 4; *breakpoints* – col. 5, lines 14-23); suitable for further analysis including compiler optimization (e.g. col. 6, lines 34-36); for interpreters (step 304 – Fig. 3), and for use in generating binary code (e.g. col. 7, lines 27-30) for the digital information appliance.

As per claim 4, Blandy further discloses tagging byte-code for performance sensitive instrumentation (e.g. Fig. 4; col. 6, lines 38-48).

As per claim 5, Blandy further discloses persistent storage of tagged sections of byte-code (e.g. Fig. 1; Fig. 2).

As per claim 7, Blandy discloses a digital information appliance for dynamic compiling, comprising a processor for implementing a program; a memory for storing program instructions (e.g. Fig. 1; Fig. 2), said program having instructions to load byte-code (e.g. *classes loader,*

Bytecodes - col. 4, lines 7-16), such byte-code being suitable for having a tagged section (e.g. *lock* – col. 4, lines 54-65; *breakpoints* – col. 5, lines 14-23); identifying the tagged section (e.g. *lock, breakpoint, branch monitor* -col. 4, line 54 to col. 5, line 23; *range, GTR* – Fig 4) and compiling the tagged section (e.g. Fig. 3; step *414* - Fig. 4); wherein the tagged section is compiled when loaded (e.g. col. 3, lines 24-28; col. 4, lines 7-16) so as to enable the processor to utilize it without additional compiling (e.g. col. 5, lines 15 to col. 6, line 33; Fig. 3, 4; *will be interpreted* – col. 6, lines 30-33).

But Blandy fails to specify that the byte code includes one dynamic base object, such object comprising an interface dynamic object and an implementation dynamic object communicating with each other over a message bus. However, this limitation has been addressed in claim 1 above, hence is rejected herein using the same rationale as set forth therein.

As per claims 8-11, these are the apparatus version of claims 2-5 above, respectively, hence incorporates the corresponding rejections of those claims respectively.

As per claim 13, Blandy discloses a system for an execution environment suitable for dynamic compilation, comprising

a memory device (e.g. Fig. 1; Fig. 2);

a loader coupled with the memory device for loading byte-code (e.g. *classes loader, Bytecodes* - col. 4, lines 7-16), such byte-code suitable for suitable for having a tagged section (e.g. *lock* – col. 4, lines 54-65; *breakpoints* – col. 5, lines 14-23); the loader being capable of interpreting(Fig. 3, 6), just in time compiling (e.g. Fig. 2, 3; col. 4, lines 7-16), and pre-compiling (Note: byte code submitted to the JIT is equivalent to having an inherent pre-compilation performed in order to have such byte-code ready for the JIT 210 of Fig. 2)

an identifier coupled with the loader for identifying the tagged section (e.g. *lock, breakpoint, branch monitor* - col. 4, line 54 to col. 5, line 23; *range, GTR* – Fig 4);

a compiler (e.g. *Compiler 210* – col. 4, lines 18-20) coupled with the identifier,

wherein the tagged section is compiled when loaded so to enable the tagged section to be utilized without additional compiling (e.g. col. 3, lines 24-28; col. 4, lines 7-16; col. 5, lines 15 to col. 6, line 33; Fig. 3, 4; *will be interpreted* – col. 6, lines 30-33).

As per claim 14, Blandy further discloses an encoding means to encode application source code to byte-code in an processor-independent form suitable for further analysis (*classes loader, Bytecodes* - col. 4, lines 7-16 – Note: the encoding means are inherent to producing of byte code form); a tagger for tagging a section of the byte-code (e.g. *lock, breakpoint, branch monitor* - col. 4, line 54 to col. 5, line 23 – Note: set up breakpoints or monitoring branching points or locks are equivalent to having a tagger to tag sections of code to interpret, skip, or generate native code therefor).

As per claims 15-16, these are the system claims corresponding to claims 3 and 4, respectively, hence incorporate herein all the corresponding rejections therein.

As per claim 18, Blandy discloses a method for providing an execution environment in an information appliance network (e.g. *PDA* - col. 3, lines 37-54), comprising

encoding an application source code in a processor-independent byte-code (e.g. *classes loader, Bytecodes* - col. 4, lines 7-16);

tagging (*lock* – col. 4, lines 54-65; *breakpoints* – col. 5, lines 14-23) at least some portion of such byte-code; and

compiling some portion of such tagged byte-code (e.g. col. 5, lines 6-23; *will be interpreted* – col. 6, lines 30-33 – Note: compiling up to some conditional branch/breakpoints is equivalent to compiling portion of code while passing over the uncompiled portions to the interpreter).

But Blandy fails to specify that the byte code includes one dynamic base object, such object comprising an interface dynamic object and an implementation dynamic object communicating with each other over a message bus. However, this limitation has been addressed in claim 1 above, hence is rejected herein using the same rationale as set forth therein.

As per claim 20, Blandy further discloses identifying of the tagged portion of byte-code (e.g. *lock, breakpoint, branch monitor* - col. 4, line 54 to col. 5, line 23; *range, GTR* – Fig 4 – Note: scope and range of bytecodes already compiled are equivalent to identifying tagged portions).

As per claims 21 and 22, Blandy discloses a bus for communication (Fig. 1) but does not mention message bus; but the message bus object limitation has been addressed in claim 1 by virtue of Coutts teaching in combination with the suggestion by Blandy. Hence this inter-processor message bus limitation (re claim 21) for communication over the network (re claim 22) would have been obvious with reference to claim 1 above for the same rationale set forth therein.

As per claim 23, see Blandy, col. 1, lines 29-49.

As per claim 24, Blandy does not discloses that the dynamic base object is programmed through a scripting language but suggests Web communication for transaction request (e.g. col. 1, lines 50-61) and a runtime environment (*Runtime data areas 204* - Fig. 2). Coutts, in a

method to request downloadable application code and data onto small appliances using JIT similar to the optimized code loading by Blandy, discloses a dynamic base object (re claim 1) and suggests use of HTML scripts to invoke report on request processing (e.g. col. 11, lines 44-49) and embedding browser technology within the Java engine to support the deployment of the transaction model (e.g. col. 36, lines 14-40). Official notice is taken that the use of HTML based scripting language, e.g. browser page and tagged documents, in browser technology and that in association with browser-based transactions was a common concept at the time the invention was made; and since Blandy's application suggests a Web based transaction using Java, the rapprochement between Java and HTML script is noted. In view of the combined suggestion by Blandy and teachings by Coutts, it would have been obvious for one of ordinary skill in the art at the time the invention was made to use the Java-written application suggested by Blandy and additionally implement it with the browser technologies such as scripting language to invoke object to assemble in the transaction as taught by Coutts because of the common practice known in Web-based client/server transaction and also of the speedy script interpretation, thus improve time and resource efficiency, e.g. less compiling time, and take full advantage of Browser given technologies and secure protocol, e.g. CGI gateway security.

As per claim 25, by virtue of the teachings of Coutts combined with the suggestions by Blandy used in the rejection of claim 1, the bi-directional limitation herein would have been obvious using the same rationale as mentioned therein (see Coutts: Figs. 14-16, 30, 32).

As per claims 26-30, these are the apparatus claims corresponding to claims 21-25 from above, respectively, hence incorporate herein all the corresponding rejections therein.

As per claim 31, this dynamic base object limitation has been addressed in claim 1 above; hence is rejected herein using the same rationale set forth therein.

As per claims 32-35, these are the system claims corresponding to claims 21, 22, 25, and 23, respectively, hence incorporate herein all the corresponding rejections therein.

As per claim 36, this claim incorporates the rejection of claim 24 above.

As per claims 37-41, these are method claims corresponding to claims 21-25 from above, respectively, hence incorporate herein all the corresponding rejections therein.

As per claim 42, Coutts discloses multiple lines of code for the dynamic base object (e.g. Fig. 41, 42, 51, 52, 57, 58 – Note: all transaction-related code to check the user and to retrieve data for display, using implementation, interface objects and message bus are equivalent to Java or C++ code statements or class objects of multiple lines). In view of the herein teaching by Coutts and the combination by Blandy/Coutts set forth in the rejection of claim 1, the multiple lines of code limitation herein would have also been obvious for the rationale set forth therein as for the dynamic base object limitation.

As per claim 43, only Coutts discloses a thin appliance (e.g. col. 38, lines 45-64) and it would have been obvious for one of ordinary skill in the art at the time the invention was made to include in Blandy's family of appliances a thin appliance as suggested by Coutts, because such appliance is even more restricted in resources and storage capacity as Blandy's suggested appliance, e.g. PDA, thus would require the same optimization schemes as disclosed by Blandy and Coutts.

As per claim 44, Blandy discloses a full threat safe (e.g. col. 54-59). In conjunction with Coutts' teachings as for the dynamic base object set forth in claim 1 above, this would also render this claim obvious by virtue of the rejection therein.

5. Claims 6, 12, 17, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blandy, USPN: 6,295,642, and Coutts et al., USPN: 6,311,165, as applied to claims 1, 7, 13, 18, and further in view of Hamby et al, USPN: 5,848,274 (art of record, hereinafter Hamby).

As per claim 6, Blandy in combination with Coutts does not specify validating that the byte-code conforms with the byte-code suitable for the application of the digital information processor/appliance. Blandy mentions about ensuring a thread safe compilation flow (e.g. col. 54-59) while Coutts suggests checking if the message received conforms with the registry of the receiving appliance (e.g. col. 44, lines 19-39). Further, Hamby, in a method to persistently allocate portions of intermediate code for optimization via incremental compilation of appliance analogous to the method of tag/skip JIT compilation/interpretation of Blandy, discloses validating that the byte-code conforms with the byte-code suitable for the application of the digital information appliance (*compiler 3000 interrogates* --- col. 27, line 42 to col. 28, line 6; Fig. 18). As suggested by Coutts or Blandy, the secure operation of downloaded or received data ought to be validated prior to or during usage of such data, and in view of such teachings, it would have been obvious for one of ordinary skill in the art at the time the invention was made to ensure that the downloaded byte code as taught by Blandy/Coutts is checked for compatibility or persistency with the operating system of the appliance (i.e. being suitable as claimed) as taught by Hamby, in order to avoid system crashes or faults during executing/using of such code,

which would considerably undermine the intent of optimizing and alleviating resources in the target appliance as taught by Blandy/Coutts.

As per claim 12, this claim the apparatus version of and corresponds to claim 6 above, hence incorporates the corresponding rejection set forth therein.

As per claim 17, this claim is the system version of and corresponds to claim 6 above, hence incorporates the corresponding rejection set forth therein.

As per claim 19, this claim includes a limitation similar the limitation of claim 6 above, hence incorporates the corresponding rejection set forth therein.

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No. 5,987,256 to Wu et al., disclosing HTML/Java rendering and displaying of objects requested with coordinate information on thin client.

7. Applicant's arguments with respect to claims 1-44 have been considered but are moot in view of the new ground(s) of rejection.

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662

Any response to this action should be mailed to:
Commissioner of Patents and Trademarks
Washington, D.C. 20231
or faxed to:
(703) 746-7239, (for formal communications intended for entry)
or: (703) 746-7240 (for informal or draft communications, please label "PROPOSED" or "DRAFT")
Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
May 16, 2003

TUAN Q. DAM
PRIMARY EXAMINER